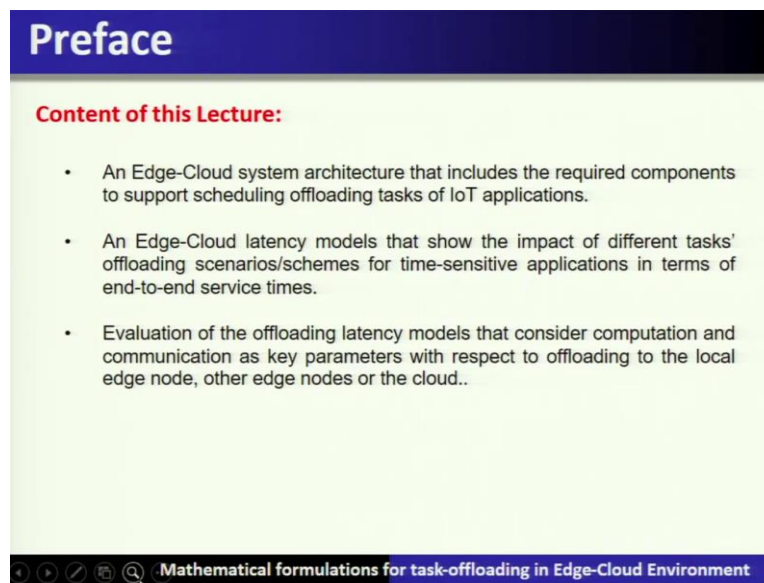


Foundation of Cloud IoT Edge ML
Professor Rajiv Misra
Department of Computer Science and Engineering
Indian Institute of Technology, Patna
Lecture 12

Mathematical formulations for task offloading in Edge Cloud

I am Doctor Rajiv Misra from IIT, Patna. The title of this lecture is Mathematical formulations for task-offloading in Edge-Cloud.

(Refer Slide Time: 00:28)



Preface

Content of this Lecture:

- An Edge-Cloud system architecture that includes the required components to support scheduling offloading tasks of IoT applications.
- An Edge-Cloud latency models that show the impact of different tasks' offloading scenarios/schemes for time-sensitive applications in terms of end-to-end service times.
- Evaluation of the offloading latency models that consider computation and communication as key parameters with respect to offloading to the local edge node, other edge nodes or the cloud..

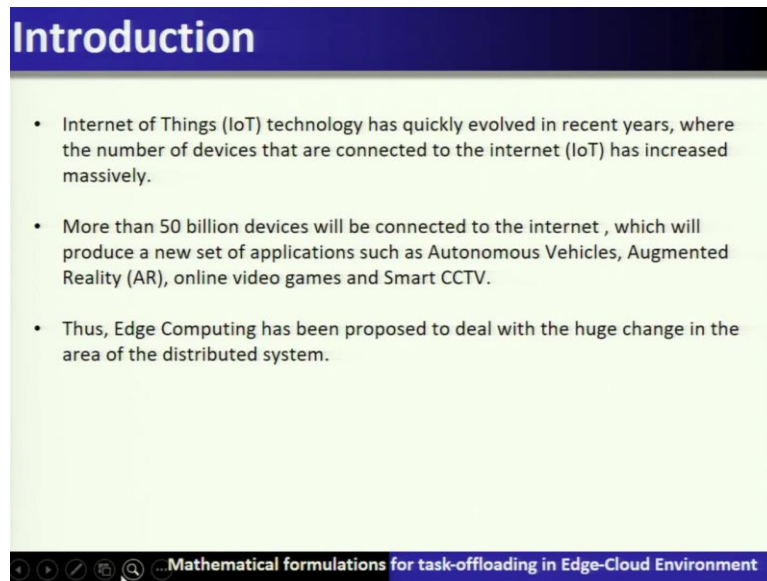
Mathematical formulations for task-offloading in Edge-Cloud Environment

Content of this lecture is as follows. So, here we are going to discuss about edge cloud system architecture that includes the required components to support shuffling of offloading task in IoT applications.

So, Edge cloud latency models for the impact of different tasks of loading scenarios and schemes for time sensitive applications in terms of end to end service times evaluation of these offloading latency models that consider computations and communication as the key parameters with respect to offloading to the local node or other edge nodes are at the cloud.

So, therefore, this architecture of an edge cloud system, how it brings the latency for time sensitive applications often, it is an IoT application scenarios, how it is going to be useful by way of scheduling offloading tasks we are going to cover in this part of the lecture.

(Refer Slide Time: 01:38)



Introduction

- Internet of Things (IoT) technology has quickly evolved in recent years, where the number of devices that are connected to the internet (IoT) has increased massively.
- More than 50 billion devices will be connected to the internet, which will produce a new set of applications such as Autonomous Vehicles, Augmented Reality (AR), online video games and Smart CCTV.
- Thus, Edge Computing has been proposed to deal with the huge change in the area of the distributed system.

Mathematical formulations for task-offloading in Edge-Cloud Environment

So, let us go to introduce you the entire concept of this particular lecture. Now, Internet of Things also known as IoT has quickly evolved in recent years. So, in IoT, what is evident is the number of devices that are connected to the internet is increasing massively more than 50 billion such IoT devices.

Which are deployed in various applications such as industry 4.0 applications or Smart City applications or some other mission critical applications will be connected to the internet and which will produce a new set of applications such as autonomous vehicles, augmented reality online video games, smart CCTV.

So, therefore, these particular devices are also called as an IoT devices which are upcoming or a new set of applications. Therefore, this particular support of collecting so many number of devices directly to the cloud has an alternative means that is called an edge computing. So, edge computing has been proposed to deal with the huge numbers in the area of distributed systems.

(Refer Slide Time: 02:58)

Motivation

- For enhancing customer experience and accelerating job execution, IoT task offloading enables mobile end devices to release heavy computation and storage to the resource-rich nodes in collaborative Edges or Clouds.
- Nevertheless, resource management at the Edge-Cloud environment is challenging because it deals with several complex factors (e.g., different characteristics of IoT applications and heterogeneity of resources).
- Different service architecture and offloading strategies quantitatively impact the end-to-end service time performance of IoT applications.
- Consequently, the latency depends on the scheduling policy of applications offloading tasks as well as where the jobs will be placed in order to meet the requirements of latency-sensitive applications.

Mathematical formulations for task-offloading in Edge-Cloud Environment

So, let us go for the motivation behind this concept, so far enhancing the customer's experience and acceleration of Java execution. This IoT task offloading enables mobile end devices to release heavy computation and storage to the resource rich nodes in collaborative edge or the cloud.

So, here what we want to say is that now, if you want to enhance the mobile devices capabilities to undertake what is their capacity beyond that, so for that, to add this customer experience or accelerating job execution, there is a concept of task offloading. So, these devices often offload their task, where there are two different resource places or centers called edge or the cloud once they offload their task, which is compute heavy or storage heavy to the edge or the cloud, which are considered as the resource rich.

So, therefore, this collaborative cloud edge model or the architecture is going to support such IoT activities such IoT devices or the mobile devices to undertake heavy computation and storage and on the other hand to have the rich or a good customer experience and activation job. So, these are the some of the important ways of supporting the IoT applications.

So, nevertheless in this particular scenario, what is important is the resource management or the edge cloud environment so that billions of IoT devices can be connected and will have with a better experience or the performance. Therefore, resource management in this particular environment that is edge cloud is a challenging because it deals with several complex factors such as the characteristics of the applications and heterogeneity of that sources.

So, this is going to be important component to support a new kind of IoT applications. And therefore, the basic concept or a notion is called the resource management in Edge cloud environment. Now, let us see that different service architecture and different offloading strategies quantitatively impact the end to end service time performance of an IoT applications.

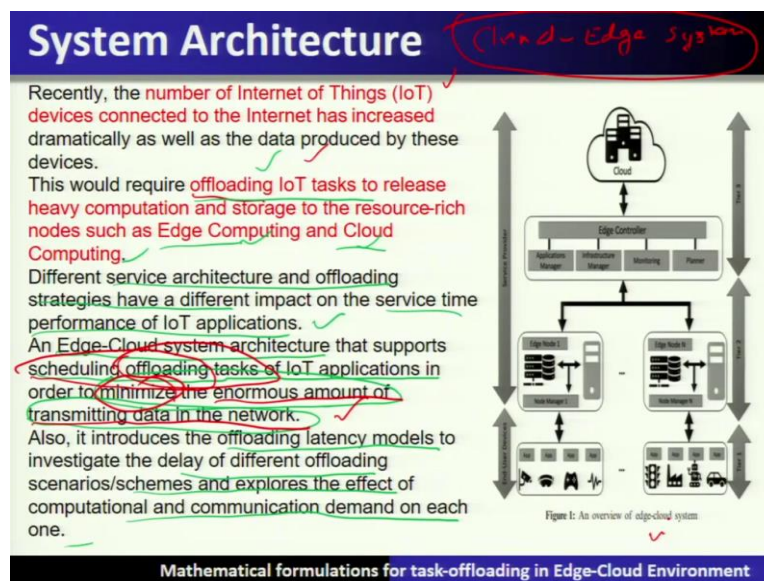
So, it is not only choosing any edge cloud environment, but you have to now see based on different service architecture, we have to now see what which kind of architecture is well suited for each cloud and varmint and how the resource management is to be done in different architectures.

And therefore, we are going to see a different task offloading strategies and different architecture and offloading strategies, these two different concepts in the edge cloud system will impact end to end service time and performance of IoT applications. So, this becomes the motivation of understanding what we mean by the Resource Management at the edge cloud environment or what is the architecture of it.

And how different offloading strategies will play into the role to have the end to end service time performance of an IoT application. Now, consequently, this latency is important factor. So, latency depends upon the scheduling policies of applications offloading the task as well as the jobs where the jobs will be replaced in order to meet the requirement of latency sensitive application.

We are going to see this notion in great details and also we will look upon through the mathematical models which of these different service architecture and which of these different offloading strategies is going to be suited for the time sensitive or end to end service time performance of an IoT application.

(Refer Slide Time: 07:02)



Let us get started with the system architecture of a cloud edge system. So, as we have pointed out, we are seeing again that the number of Internet of Things that is IoT devices, which are connected to the Internet has increased dramatically, as well as the data produced by these devices are also increasing day by day. And therefore, support from the cloud is very much needed to manage or to evolve this era of billions of IoT devices, which is useful for the automation.

Therefore, this offloading of an IoT tasks which billions of IoT devices are generating this offloading the IoT task to release the heavy computation and storage of a resource rich nodes such as edge and the cloud system is needed. So, therefore, IoT devices are not that resource rich devices, they are called resource constrained devices, but they undertake a heavy computation and storage type of applications.

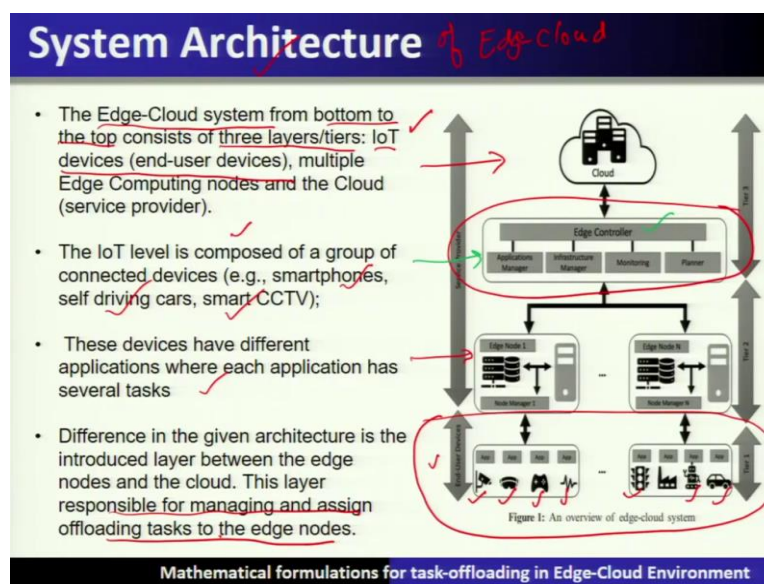
Therefore, offloading these IoT tasks to the edge or to the cloud is very important notion to support these applications running on these IoT devices. Therefore, different service architecture and offloading strategies will have different impact on the service time performance of an IoT application. Let us understand through this edge IoT system architecture that supports shuffling, offloading task of IoT applications, with the objective to minimize the enormous amount of data which are transmitted in the network.

So, this introduces the offloading latency models to investigate the delay of different offloading scenarios and exploit the effect of computational and communication and demand in each one of them. That means, let us summarize what do we mean by this particular

architecture that can support this shearing offloading or offloading tasks of an IoT applications with the objective to minimize this enormous amount of transmission into the network?

So, what are we going to minimize is that how are we going to schedule the task offloading strategies in under which architecture to support this IoT application, whereas with the objective to be demise, the enormous amount of data transmission on the network? So how that is all possible is through this important architecture which we are going to explain here in this particular lecture.

(Refer Slide Time: 09:56)



So, let us get started. Add with that edge IoT architecture to support this IoT task offloading and scheduling for different resource management. So, this IoT system, we are going to explain the architecture of IT architecture of edge cloud system to explain this architecture of a edge cloud system, we consider a three layer or three tier model and we will explain from bottom to the top manner. So, at the bottom layer, what you will find is and IoT devices.

So, for example, here you can see that is the key or one or the bottom most layer will contain the end user devices. So, example of end user devices are the cameras or the augmented reality headsets, different applications or road side signals or different apps or self-driving car, they are all IoT devices, if they are connected and following the principles of Internet of Things. So, they are also called the end user devices.

So, these IoT level so this is the level one or Tier one is composed of a group of connected devices such as smartphones, self-driving car smart CCTV and so on. Now, these devices

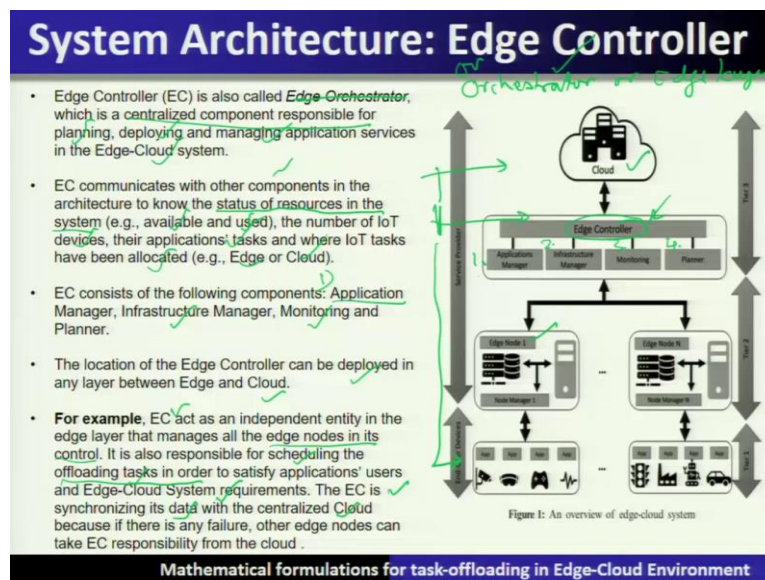
have different applications where each application is running different tasks. So, you can see that there are a diverse set of IoT devices, which are to be supported by a common architecture of an edge cloud system.

So, this becomes the layer one of an edge cloud system architecture and that is the bottom most layer. Now, the difference is in the given architecture is introduced layers between the edge nodes and the cloud this layer is responsible for managing and assigning the offloading task to the edge nodes.

So, meaning to say that, if this is the edge node and this is the cloud, then in between there is a layer there is a layer of management, this particular layer of management is responsible for managing and assigning the offloading tasks to the edge nodes. So, now, the question is resolved in the sense that who takes care of this resource management of IoT devices task offloading, so it is this particular layer that is called Edge layer.

So, Edge layer is also known as edge controller orchestrator, it has different names, let us get started to know more detail about this important component that is called the edge layer.

(Refer Slide Time: 12:57)



So, the edge layer is sometimes called we call it as edge controller or it is sometimes called as orchestrator or it is called Edge layer. So, this edge controller sits between the cloud and the edge, this is the edge so it sits between the cloud and edge. So, Edge controller is also called Edge orchestrator and which is the centralized component responsible for planning, deploying and managing application services in the edge cloud system.

So, to exploit this edge cloud system by an IoT this particular edge controller or orchestrator is responsible or it will facilitate now, this edge controller will communicate with other components in this architecture. That means, this particular edge controllers will now communicate with the cloud and the edge nodes.

These are different other components and an IoT devices are different components in this architecture to collect the status of the resources in the system. For example, whether what system resources are available or what system resources are already in use, what are the number of IoT devices running in that applications running their applications task.

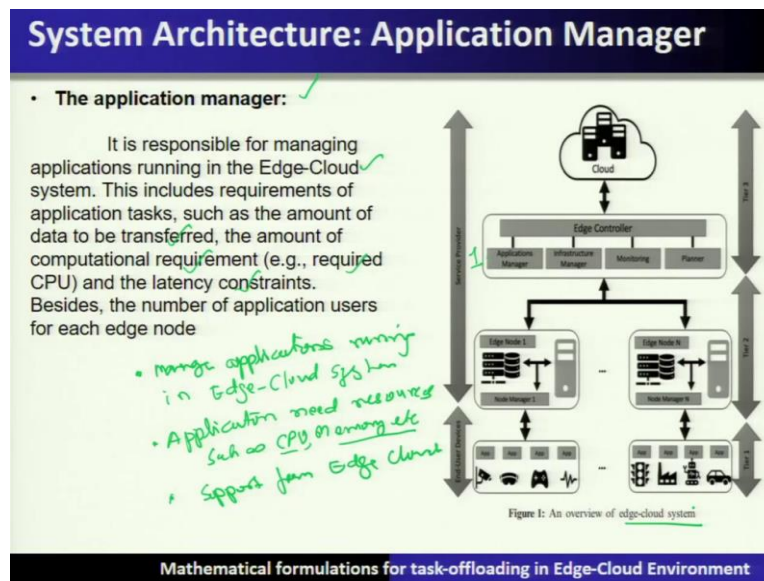
And where these IoT devices and where these IoT tasks have been allocated whether on the edge or on the cloud, all these statistics are the information will be now collected by the edge controllers by way of communicating with all these components. So, therefore, Edge controller consists of following component to understand the functionality of edge controller, we have to see that the edge controller has following components to do this task.

The first one is called Application Manager. The second one is called infrastructure manager. Third one is called the monitoring. And fourth one is called a planet, let us understand one by one so that we can understand the functionality of edge controller, the location of the edge controller can be deployed in any layer between edge and cloud that we have already explained. Now, for example, Edge controller acts as an independent entity in the edge layer that manages all the edge nodes in its control.

So, these are all you can see that below the cloud, there are edge nodes to manage these edge nodes on and the communication between the cloud the edge controller or the edge layer sits in between. So, it is responsible for scheduling the offloading task in order to satisfy application users and edge cloud system architecture. Now, it has two roles on one side it has to now satisfy the requirements of an IoT devices or end user devices.

And on the other hand, it has to now manage the resources of cloud system for providing these resources in the form of task allocation or scheduling the offloading tasks of an applications which are running by the users of IoT devices. Therefore, Edge controller is synchronizing its data with a centralized cloud, because if there is any failure, that is other edge nodes can take the responsibility of the edge controller responsibility from the cloud. So, these are some of the important applications.

(Refer Slide Time: 16:31)

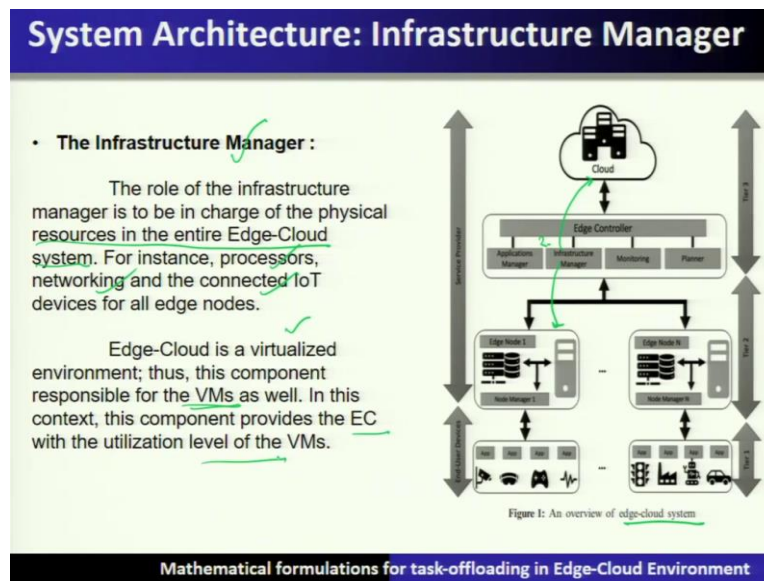


Now, let us understand the details of the competence of the edge controller the first one is called Application Manager. So, Application Manager is responsible for managing the applications running the edge cloud system, this includes the requirement of application tasks such as amount of data to be transmitted, and the amount of computational resources which are required such as CPU, memory and so on, and what are the different latency constraints.

So, besides the number of application users for each of the nodes, so let us understand this application manager. So, important component of the application manager is to manage applications running in the edge cloud system meaning to say that these applications needs the resources such as CPU, memory etcetera.

Now, to support these resources from edge cloud system So, this particular application manager has to manage these applications running so that it can get the support from this edge cloud system. So, this is the first task of an Application Manager.

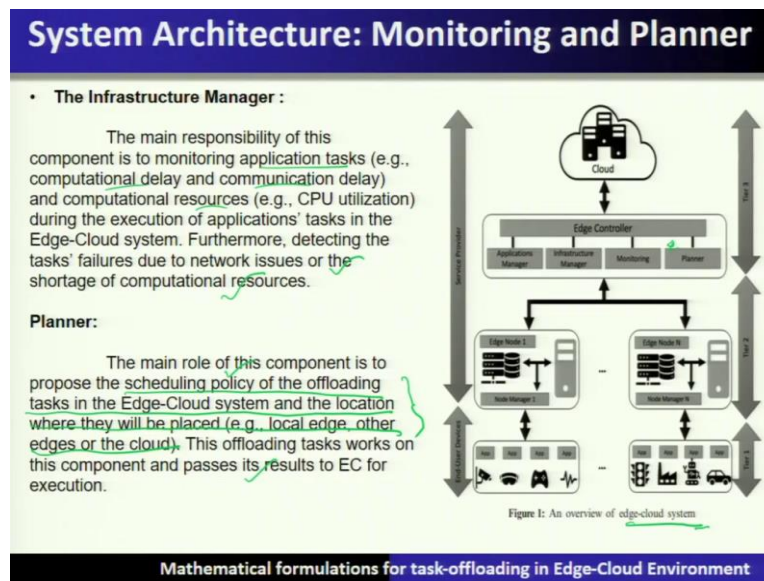
(Refer Slide Time: 18:17)



Second important task of the app is called infrastructure manager the role of the infrastructure manager is in charge of the physical resources of the entire edge cloud system. So, this infrastructure manager now has the responsibility to collect all the information about the resources of edge cloud system that is for instance, the processors networking and different connected IoT devices.

So, all the resources whatever is available in Edge cloud system in Protection Manager used to collect and inform. So, Edge cloud is in an virtualized environment, when we talk about the resources they are packaged into a unit which is called virtual machines. So, in this context, this component provides the edge controller with a utilization level of virtual machines.

(Refer Slide Time: 19:12)

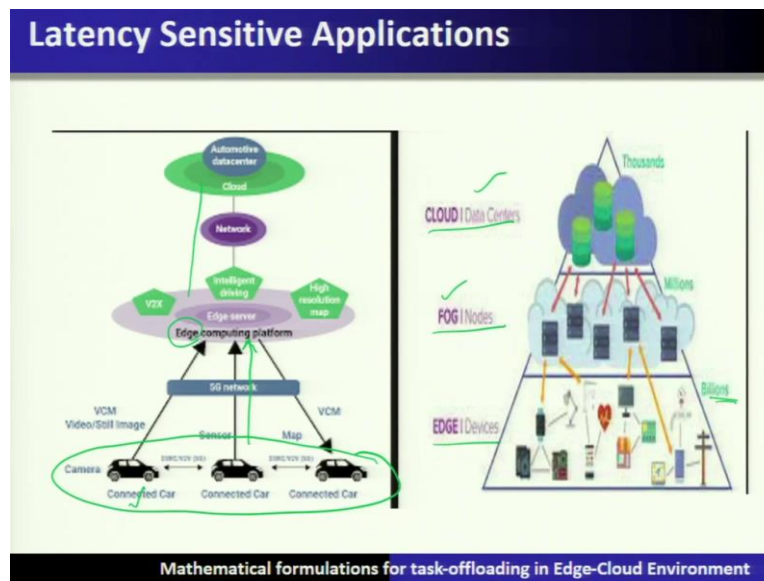


So, therefore, main responsibility of this infrastructure manager is to monitoring the application task computational delay and communication delay and computational resources during the execution of the application task. And also it will be involved in detecting the task failures due to the network issues are shortages of computing resources.

Next important component of a task edge controller is the planner. The main role of this planner is to propose a scheduling policy of the offloading task and edge cloud system and the location where they are going to be placed. So, therefore, this planner after getting the complete picture of this availability of the resources, which are available.

And then what are the requirements of an IoT system applications, this planner will formulate some scheduling policy of offloading the task in the edge cloud system, this offloading task works on this component passes its result to the edge controller for the execution.

(Refer Slide Time: 20:17)



So, let us see about the latency sensitive application how they are supported and in the edge cloud system. So, you can see that connected cars are you can see these are the devices or IoT devices, they often connect to the cloud earlier days, but now, they are going to be connected because they are latency sensitive. So, they will be connected very close to the device they are called edge.

So, edge nodes or XR words are the means to provide latency sensitive to support latency sensitive application, their resource requirements in a latency sensitive manner is going to be supported. And you know that these ad systems also has the limitations in compared to the cloud. Therefore, this edge system is often connected to the cloud.

So, whatever more resources which is not possible to be allocated to from the edge will be now coming from the cloud. Therefore, sometimes it has these different components which are involved in the latency sensitive applications at the edge you will find various devices and these edge devices will be there and which are going to support an IoT devices and above that on the network layer.

The devices having this compute, memory and storage and network capacities also can undertake some form of task offloading from these edge or an IoT devices called fog nodes. And above that is called the cloud which is in the form of data centers.

(Refer Slide Time: 21:54)

Latency Sensitive Applications

- Latency-sensitive applications have high sensitivity to any delays accrue in communication or computation during the interaction with the Edge-Cloud system.
- For instance, the IoT device sends data to the point that processing is complete at the edge node or the cloud in the back end of the network, and the subsequent communications are produced by the network in response to receive the results.
- For example, self-driving cars consist of several services, classified these services in categories based on their latency-sensitivity, quality constraints and workload profile (required communication and computation).
 - First, critical applications, which must be processed in the car's computational resources, for instance, autonomous driving and road safety applications.
 - Second, high-priority applications, which can be offloaded but with minimum latency, such as image aided navigation, parking navigation system and traffic control.
 - Third, low-priority applications, which can be offloaded and not vital as high-priority applications (e.g., infotainment, multimedia, and speech processing).

Mathematical formulations for task-offloading in Edge-Cloud Environment

- rich customer experience

Now, let us talk about latency sensitive applications. So, latency sensitive applications, how high sensitivity to any delays that occur in the application, for example, self-driving car, when it is cameras can see the outside view now, what to move whether the car should move forward or at what speed all these decisions has to be taken based on the camera feeds of an outside environment. So, this is highly sensitive to any delays.

Now, if such data is sent to the cloud for doing the computation, then it will take enormous amount of time. And those kinds of that is the latency or the delays cannot tolerate and therefore they are called latency sensitive applications. So, latency sensitive applications, again, I am repeating have the highest sensitivity to any delay accrue in the communication or the confusion during interaction with Edge cloud system.

For example, the IoT device sends the data to the point that processing is complete at the edge node or at the cloud in the backend of the network and subsequently the communications are produced by the network in response to receive so therefore, there are two activities very important in this edge cloud system to support this latency sensitive application.

The first is called communication latency, the latency due to the delays due to the communication the latency due to the computation. So, both these latencies are to be reduced to a minimal to support this latency sensitive applications. For example, self-driving car consists of several services and classified these services in the category based on latency sensitivity and the quality constraint and workload per file.

Therefore, it requires the communication and the computation is to be done at a very minimal latency or the delays in such critical applications. So, these critical application must be processed in the cars computing resources that is at the IoT devices add if they do not have those amount of resources, it will lease from the edge or from the cloud to have this autonomous driving on the road with a full safety compliance.

The second-high priority applications which can be offloaded to the edge cloud system, but with a minimum latency such as image edit navigation, parking, navigation system traffic control and so on third low priority applications which can be offloaded but not vital as High Priority applicants such as infotainment multimedia in speech to have this rich customer experience.

So, those kinds of applications are also latency sensitive applications, but they are not critical compared to the first two categories. Therefore, offloading also is very much important to support rich customer experience in the applicant’s instances infotainment that is sitting in the car and getting all the video or audio contents, multimedia contents, multimedia speech processing and so on.

(Refer Slide Time: 25:01)

Latency Sensitive Applications	
Latency-sensitive applications	
Industry	Applications
Industrial automation ✓	<ul style="list-style-type: none"> ✓ Industrial Control ✓ Robot Control ✓ Process Control
Healthcare Industry ✓	<ul style="list-style-type: none"> ✓ Remote Diagnosis ✓ Emergency Response ✓ Remote Surgery
Entertainment Industry ✓	<ul style="list-style-type: none"> ✓ Immersive Entertainment ✓ Online Gaming
Transport Industry ✓	<ul style="list-style-type: none"> ✓ Driver Assistance Applications ✓ Autonomous Driving ✓ Traffic Management
Manufacturing Industry ✓	<ul style="list-style-type: none"> ✓ Motion Control ✓ Remote Control ✓ AR and VR Applications

Mathematical formulations for task-offloading in Edge-Cloud Environment

So, let us see the latency sensitive application are categorized in the following domains in the industry segments they are called industrial automation. So, the industrial the latency sensitive application in the industrial automations are industrial control robot control process control, whereas for the healthcare industry, these latency sensitive applications are removed diagnosis emergency response remote surgery for entertainment industry.

These latency sensitive applications are immersive entertainment, online gaming and for transport industry such latency sensitive applications or driver assistant applications autonomous driving traffic management for manufacturing industry, it is motion control remote control, AR and VR applications.

So, these are all latency sensitive applications. Also there are some new some are existing applications which are going to be supported to support this latency sensitive it requires architecture of an edge cloud system and the resources or location becomes very very important in this case.

(Refer Slide Time: 26:06)

Latency Models

- Modelling the various offloading decisions for IoT tasks that can increase the Quality of Service (QoS).
- With the increasing number of IoT devices, the amount of produced data, the need for an autonomous system that requires a real-time interaction as well as the lack of support from the central Cloud due to network issues, service time has been considered as one of the most important factors to be handled in Edge Computing.
- One of the main characteristics of Edge Computing is to reduce the latency level.
- Additionally, using Edge Computing will enhance application performance in terms of overall service time comparing to the traditional Cloud system.
- However, different offloading decisions within the Edge-Cloud system can lead to various service time due to the computational resources and communications types. The current real-world applications measure the latency between the telecommunication service provider and the cloud services.
- Compare the latency between offloading to the edge or the cloud, latency between multiple edge nodes that work collectively to process the offloading tasks. Investigating the latency of the Edge-Cloud system is an essential step towards developing an effective scheduling policy.
- Firstly, task allocation in the Edge-Cloud system is not only two choices (e.g., either at IoT device or in the cloud), but could be on any edge nodes. Moreover, edge nodes connected in a loosely coupled way on heterogeneous wireless networks (i.e., WLAN, MAN and WAN), making the process of resource management and the offloading decision more sophisticated.
- Secondly, given that task processing is allocated among multiple edge nodes working collectively and the cloud, it is challenging to make an optimal offloading decision. The latency models to investigate the delay of different offloading scenarios/schemes.

Mathematical formulations for task-offloading in Edge-Cloud Environment

So, let us understand what is this latency models available to support these latency sensitive application. So, modeling various offloading decisions for the IoT task can increase the quality of service. Now, with increasing the number of devices IoT devices, the amount of data produced and the need for autonomous system requires the real time interaction as well as the lack of support from the cloud due to the network issues.

So, service time has become one of the most important factor to be handled by the edge computing which was earlier happening at the cloud, but what is not supporting that latency sensitive application. So, one of the main characteristics of edge computing is to reduce the latency level. So, this is what is the important driving factor of edge computing. And therefore, we are discussing about this architecture and resource allocation and the task offloading strategies.

So, therefore, edge computing will enhance the application performance of the overall service time compared to the traditional cloud system. Now, different offloading decisions within the edge cloud system can lead to various service time and due to the computational resources and communication types, so the current word applications measure the latency between the telecommunication service provider and the cloud provider.

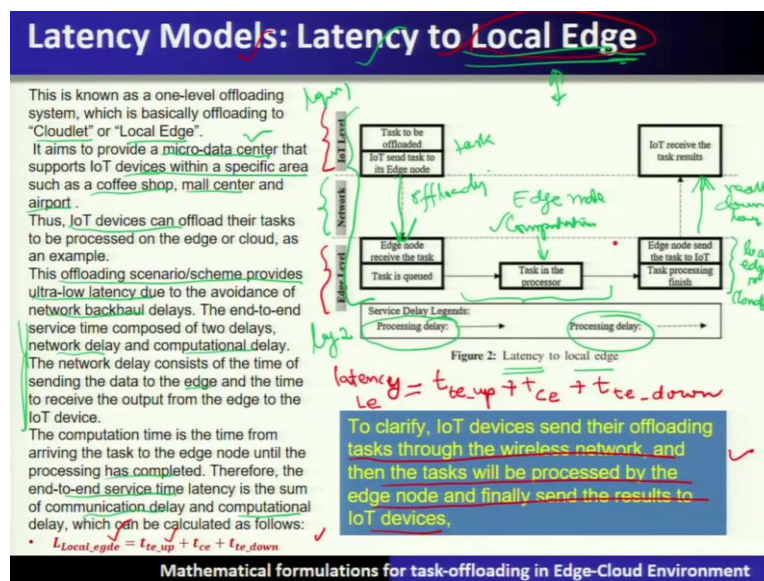
So, cloud provider provides the computation and service provider telecom service for a communication these two latencies are going to be important factor in modeling this latency into the scenario. So, to compare the latency between the offloading to the edge or the cloud, the latency between multiple edge nodes that work collectively to process the offloading task. So, we are going to investigate here in this way using our data latency model.

How the latency model can be applicable for the edge cloud system AI and towards developing effective scheduling policy after understanding this particular modeling of latency in this scenario, that is edge cloud system them. So, the task of floating in Edge cloud system is not only having the choices that is whether the task can be done at IoT devices or at the clouds, but could be on any of these edge nodes as well.

So, therefore, it requires a process of resource management and offloading decisions and also in a very sophisticated way. Secondly, the task processing is allocated across multiple edge nodes working collectively in the cloud and challenging to make this optimal offloading decisions. So, before going ahead at the optimization levels, we have to understand how to model using some of the latency models.

So, we will explore one such latency model in which we will be able to analyze or investigate the latency sensitivity, which is based on the offloading decisions.

(Refer Slide Time: 29:14)



So, let us get started with the latency models. So, the first latency model which we are going to consider is only hurting the local edge. So, you can see that last two layers we are going to consider here under this model, which is very simplified model to understand this particular latency. So, it contains the IoT layer and the edge layer and to connect IoT with edge, so we have to now also consider the network in between.

So, let us see and this model we are calling as local edge. So, this is known as one level offloading system and it is basically offloading to the cloud layer or the local edge. So, this is sometimes called as the local edge or it is also called a cloud lit. Now, this particular local edge or a cloud led, often is a micro data center.

So, micro data center means it is having a massive data center, but a small form of that data center is at the edge and it is in the form of edge nodes or a local edge is in the form of micro data centers that support IoT applications within a specific areas. For example, if you go to a coffee shop, the offloading of these tasks task offloading from an IoT from coffee shop or the mall center or an airport does not require the complete cloud.

But it requires a micro data center. And this is an example of a local data center and we have to now understand in that scenario, what is the effect of latency how we can reduce the latency. So, IoT devices can offload the task to be processed on the edge or the cloud, this offloading scenario provides ultra-low latency due to the avoidance of long-haul network delays. So, you can see that this particular network is a very short-range communication and it does not require a very long-haul communication.

So, network delay is going to be very minimal and also very close by the very near to the devices is the edge nodes in the form of micro data center, they are for that particular network delay is very minimal. Similarly, this particular micro data centers are also very sophisticated. Therefore, it will also reduce to the communication delay. So, network delays consist of the time of sending the data to the edge nodes.

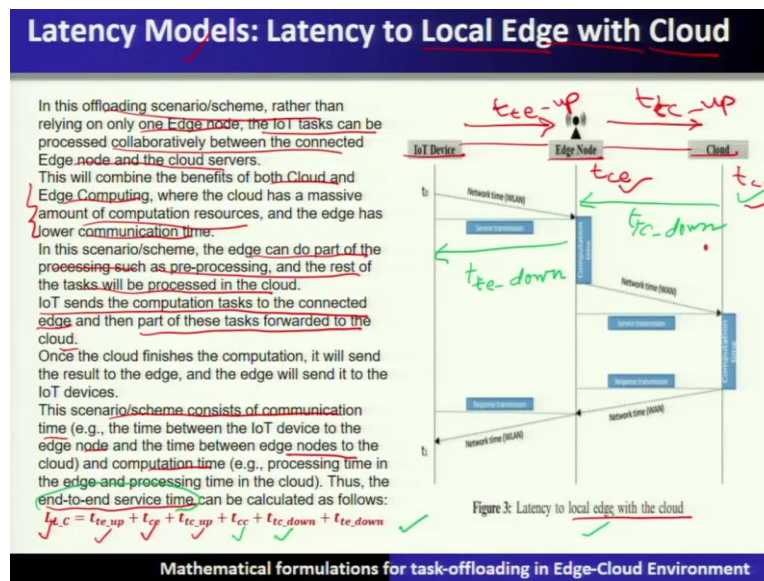
So, that is what is shown over here. So, IoT will send the data to the edge nodes sending the data to the edge and the time to receive this output. So, once it sends to the edge nodes, so as node will now involve is micro data center for doing the computation, this is called computation at the edge and then the results will again be sent back. So, this is called offloading sometimes often called as a task offloading and this is called downloading what is downloading the results are downloading.

So, therefore, if you if you measure the latency in this local edge environment, so the computation time is the time from arriving the task to the edge node until the processing is complete. Therefore, if you look at the end to end service time latency, this is the sum of the communication delay plus computational delay. So, this is called Communication delay and this is called computation delay.

So, here you can see that this is having two types of delay and using this particular equation which we are going to explain will match In this latency so latency formula, you can see that at the local edge will be goes like this. So, the time for IoT to the edge uploading. So, the uploading that is called time from IoT to the edge uploading plus the time to do the computation at the edge, so time to do the computation at the edge plus the time to download the results from the edge to the IoT devices will be the latency of the local edge.

So, that is what is written over here. Now, to clarify that IoT device sent, they are offloading tasks through the wireless network and then the task will be processed by the edge node and finally, send the result to the IoT devices is the latency calculation in the local edge scenario.

(Refer Slide Time: 34:37)



Now, let us see that if we extend the model that is the local edge with the cloud, so you can see that IoT device and then comes the edge nodes and then we are also using the cloud. So, we have now complicated the model that is the local edge be the cloud. So, let us see the latency and the offloading scenarios here in this case, this offloading scenario is key rather than relying on only one edge node, the IoT task can be processed collaboratively between the connected edge node and the cloud servers.

Now, this particular collaboration will bring into more such resources, which was earlier the constraint at the edge node will be able to be provided by the cloud therefore, different kinds of applications are going to be supported in this particular environment. This will combine the benefit of both cloud and the edge computing. So, cloud you know that it has the massive amount of computation resources and the edge has the lower complication time, but if you combine you will support a new kind of IoT application.

So, in this scenario, the edge can do a part of the processing here, such as preprocessing and rest of the task will be processed at the cloud. So, both collaboratively edge and the cloud together will be used to offload the task from IoT and to solve the problem in a latency sensitive manner. Therefore, the IoT sends the computation task to the connected edge. So, it will be sent fast and then part of these tasks will be forwarded.

So, partially it will be forwarded to the cloud. So, edge node will forward some of the task to the cloud to complete that. So, this scenario consists of communication time, that is the time between the IoT to the edge node and the time between edge node to the So, this is the

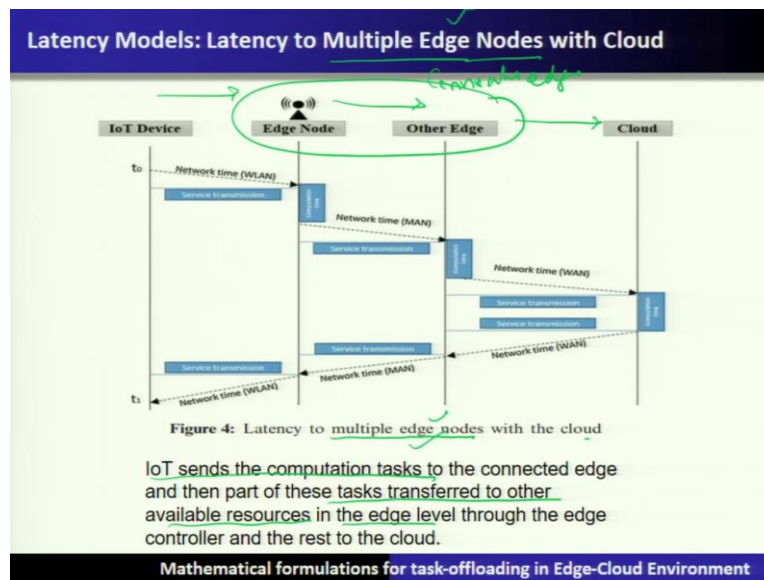
communication time that is the time from IoT to the edge node and the time from edge node to the cloud.

These are the two different communication delay and computation time that is the processing at the edge. So, this is the computation time computing at the edge and then the computing and the processing time and the processing time computing at the cloud PCC. So, if you sum them the total end to end service time, which is nothing but the latency in this model will be calculated as follows.

So, latency in a local edge with the cloud is nothing but the time from IoT devices to the edge node in the uploading the computation at the edge node plus time for the edge node to upload the computation to the cloud that is PC upload and then the configuration at the cloud in the configuration at the cloud that is TCC is written and then the downloading So, the results of computation from the cloud to the edge node that is TC downloading.

And finally, the results of the confusion the edge node also has to be downloaded to the IoT devices. If you sum them, it brings out to be end to end service time latency and this formula has explained all that details.

(Refer Slide Time: 38:13)



Now, we are changing the model to include multiple edge nodes with the cloud. So, earlier we have included only one edge node, but now we see that there exist other edge nodes also as an opportunity to do the computation. Therefore, this model is using the multiple edge node with the cloud. So, let us explain this particular model in more details. So, here you can see that the IoT sends the configuration task to the connected.

So, this is called a connected edge node and then part of these tasks is transferred to the other available resources in the edge level through the edge controller and rest and rest to the cloud. So, Edge controller will manage the resources of the connected edge. So, therefore, IoT tasks can offload to the edge node and from the edge node to the other edge node, and the remaining part of the computation will be offloaded to the cloud for doing the computation. So, this becomes a model which is called a multiple edge nodes with the cloud.

(Refer Slide Time: 39:25)

Latency Models: Latency to Multiple Edge Nodes with Cloud

- This is known as a three-level offloading scenario/scheme that aims to utilize more resources at the edge layer and support the IoT devices in order to reduce the overall service time.
- It adds another level by considering other available computation resources in the edge layer.
- Basically, it distributes IoT tasks over three levels: connected edge, other available edge nodes and the cloud.
- The edge controller (edge orchestrator) controllers all edge servers by Wireless Local Area Network (WLAN) or Metropolitan Area Network (MAN), which have low latency compared to Wild Area Network (WAN).
- This will help to decrease the dependency of cloud processing as well as increase the utilization of computing resources at the edge.
- This scenario/scheme consists of communication time (e.g., the time between the IoT device to the edge node, the time between edge node to other collaborative edge node and the time between edge nodes to the cloud) and computation time (e.g., processing time in the edge, processing time in other collaborative edge node and processing time in the cloud). Thus, the end-to-end service time can be calculated as follows:

$$L_{three_off} = t_{te_up} + t_{ce} + t_{ceo} + t_{tc_up} + t_{cc} + t_{tc_down} + t_{teo_down} + t_{te_down}$$

Mathematical formulations for task-offloading in Edge-Cloud Environment

So, let us see the latency calculation. So, this is known as three level offloading scenario and aim is to utilize more resources at the edge layer and then support the IoT devices in order to reduce the overall service time. So, if more resources are available from the edge is only possible with a multiple edge node that is a connected edge.

So, this is a new model which will still bring the latency to a very minimal so and dependency from the cloud is also reused and therefore, the latency that is communication latency is also reduced in this model. So, this adds up another layer by considering other available computational resources at the edge layer.

So, it distributes the IoT task over three levels connected edge and other available edge nodes and the cloud. So, who does this responsibility of knowing what are the resources available at the connected edge is called the edge controller orchestrator controllers has all the edge servers which are connected by the local area network or a metropolitan area network and have a low latency compared to that wide area network.

So, let us see that in this particular scenario, what is the communication time So, communication time consists of the time between the IoT devices to the edge nodes time between the edge node to the other collaborative edge nodes and the time between the edge node to the cloud. So, this comprises of the time between the IoT device to the edge nodes and then this particular time for other edge nodes then the time for uploading to the cloud and then the time to download from the cloud to the edge nodes or to the other edge nodes and from other edge nodes downloading.

And then finally, from the edge nodes downloading it besides this communication delays, there are two delays for the computation one is the computation at the edge computation on the edge and then computation at the other edge, and the computation at the cloud. So, you can see that the availability of other edge.

The compute computation time is reduced in comparison to the doing the computation in the cloud. So, this particular important addition of other edge or multiple edge nodes will bring down the latency as compared to doing the computation at the cloud. So, therefore, this is an important development.

(Refer Slide Time: 42:15)

Experiment

Assumptions:

- We have three edge nodes connected to the cloud.
- Each edge node has two servers, and each of them has four VMs with a similar configuration.
- The cloud contains an unlimited number of computational resources

Key parameters of the simulation environment :

Key parameters: Values

- Simulation Time :30 min ✓
- Warm-up Period :3 min
- Number of Iterations: 5
- Number of IoT Devices: 100-1000 ✓
- Number of Edge Nodes :3 ✓
- Number of VM per Edge Server: 8 ✓
- Number of VM in the Cloud :not limited ✓
- Average Data Size for Upload/Download (KB) :500/500 ✓

Mathematical formulations for task-offloading in Edge-Cloud Environment

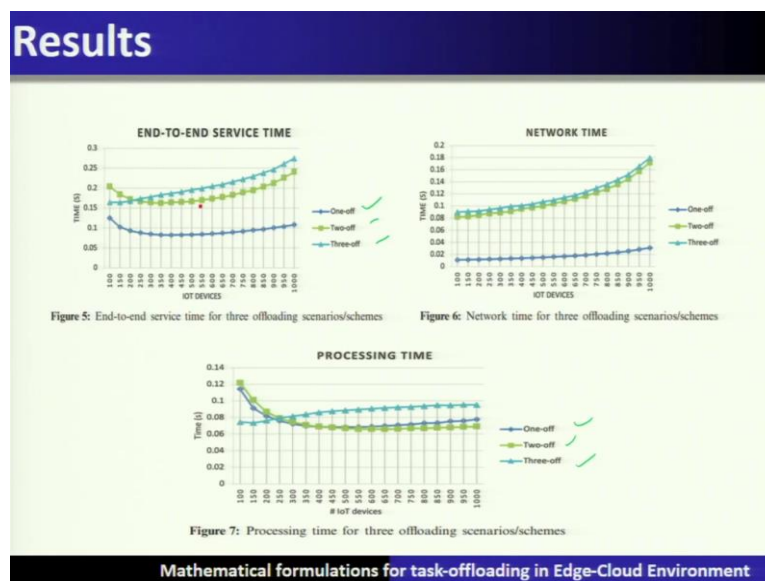
And let us see some of the experiments how you can evaluate various different type of architecture under this age cloud system. For that you require several nodes which are to be connected you have to assume to do the experiments nodes which are connected to the cloud and you have to assume the availability of different virtual machines and also different computational resources which are available for the scheduling or the task offloading.

So, these are some of the important things that let us say that this experiment has considered 100 to 1000 IoT devices and the number of edge nodes are three and this particular virtual machines at the edge is eight and the virtual machines at the cloud is not limited and average data size for uploading and downloading is 500.

So, let us understand here in this case, you have below layers IoT layers, let us say that you have 1000 devices which are connected to three edge nodes which are connected edge nodes and then you have the cloud. Now, the data size of offloading that is the offloading of the task that is the communication is happening is assumed that it is of the 500 kilobytes this is one unit. Now, here we are also considering the communication between the edge nodes is also in the range of 500 kilobytes.

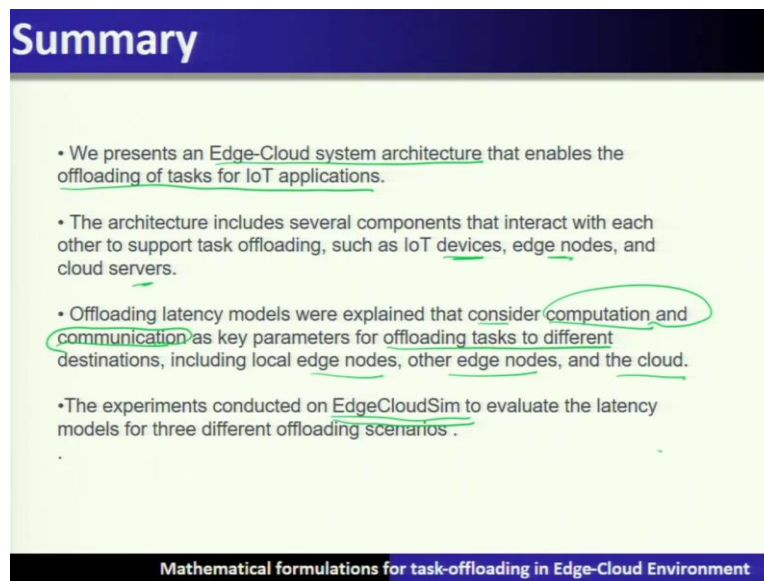
Now, this particular architecture is used to simulate using different servers. For example, here there are two servers, we assume in every edge node. And each of them has four virtual machines. Each one of them has four virtual machines. And this IoT will now do the task offloading and you can now consider using this particular simulation model.

(Refer Slide Time: 44:42)



To do this, and this will bring up some of the results of an IoT device. So, you can see under different models' architectures, we have now discussed all these things.

(Refer Slide Time: 44:57)



Summary

- We presents an Edge-Cloud system architecture that enables the offloading of tasks for IoT applications.
- The architecture includes several components that interact with each other to support task offloading, such as IoT devices, edge nodes, and cloud servers.
- Offloading latency models were explained that consider computation and communication as key parameters for offloading tasks to different destinations, including local edge nodes, other edge nodes, and the cloud.
- The experiments conducted on EdgeCloudSim to evaluate the latency models for three different offloading scenarios .

Mathematical formulations for task-offloading in Edge-Cloud Environment

So, let us summarize that we presented here, the edge cloud system architecture to support the offloading of the task in the IoT scenario. So, that is architecture includes several components that interact with each other to support the task offloading such as IoT devices, edge nodes and the cloud servers.

So, offloading latency models were explained here in this lecture, considering the computation and the communication as the key factors which are contributing to the delays and we will measure these computation and communication latency in the task offloading under different destinations including local edge and other edge nodes and the cloud.

We have also shown that what are the different experiment models so that you can conduct such experiments on any simulator, such as edge cloud simulator to evaluate the latency model under these different offloading scenarios. Thank you.